Injecting the score of the first-stage retriever as text improves BERT-based re-rankers

Askari, A.; Abolghasemi, M.A.; Pasi, G.; Kraaij, W.; Verberne, S.

Discover Computing

Research

Injecting the score of the first-stage retriever as text improves BERT-based re-rankers

Arian Askari[1] · **Amin Abolghasemi**[1] · **Gabriella Pasi**[2] · **Wessel Kraaij**[1] · **Suzan Verberne**[1]

Received: 29 September 2023 / Accepted: 23 April 2024
Published online: 26 June 2024

Abstract
In this paper we propose a novel approach for combining first-stage lexical retrieval models and Transformer-based re-rankers: we inject the relevance score of the lexical model as a token into the input of the cross-encoder re-ranker. It was shown in prior work that interpolation between the relevance score of lexical and Bidirectional Encoder Representations from Transformers (BERT) based re-rankers may not consistently result in higher effectiveness. Our idea is motivated by the finding that BERT models can capture numeric information. We compare several representations of the Best Match 25 (BM25) and Dense Passage Retrieval (DPR) scores and inject them as text in the input of four different cross-encoders. Since knowledge distillation, i.e., teacher-student training, proved to be highly effective for cross-encoder re-rankers, we additionally analyze the effect of injecting the relevance score into the student model while training the model by three larger teacher models. Evaluation on the MSMARCO Passage collection and the TREC DL collections shows that the proposed method significantly improves over all cross-encoder re-rankers as well as the common interpolation methods. We show that the improvement is consistent for all query types. We also find an improvement in exact matching capabilities over both the first-stage rankers and the cross-encoders. Our findings indicate that cross-encoder re-rankers can efficiently be improved without additional computational burden or extra steps in the pipeline by adding the output of the first-stage ranker to the model input. This effect is robust for different models and query types.

Keywords Injecting BM25 · Two-stage retrieval · Transformer-based rankers · BM25 · DPR · Combining lexical and neural rankers

1 Introduction

A commonly used ranking pipeline consists of a first-stage retriever, e.g. BM25 [2], that efficiently retrieves a set of documents from the full document collection, followed by one or more re-rankers [3, 4] that improve the initial ranking. An effective reranking strategy are BERT-based models with a cross-encoder architecture, concatenating the query and the candidate document in the input [4–7]. In this paper, we refer to these re-rankers as Cross-Encoder$_{CAT}$ (CE_{CAT}). In the common re-ranking set-up, BM25 [2] is widely leveraged [8–10] for finding the top-*k* documents to be re-ranked; however, the relevance score produced by BM25 based on exact lexical matching is not explicitly taken into account in the second stage. Besides, although cross-encoder re-rankers substantially improve the retrieval effectiveness compared

✉ Arian Askari, a.askari@liacs.leidenuniv.nl; ✉ Suzan Verberne, s.verberne@liacs.leidenuniv.nl; Amin Abolghasemi, m.a.abolghasemi@liacs.leidenuniv.nl; Gabriella Pasi, gabriella.pasi@unimib.it; Wessel Kraaij, w.kraaij@liacs.leidenuniv.nl | [1]Leiden Institute of Advanced Computer Science, Leiden University, Leiden, Netherlands. [2]Department of Informatics, Systems and Communication, University of Milano-Bicocca, Milan, Italy.

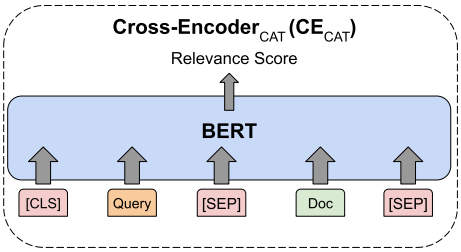

Fig. 1 Regular cross-encoder input

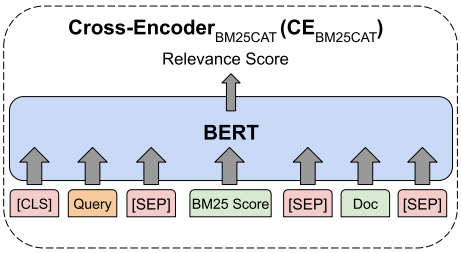

Fig. 2 Injection of BM25 in input

to BM25 alone [11], Rau et al. [12] show that BM25 is a more effective *exact lexical matcher* than CE_{CAT} rankers; in their exact-matching experiment they only use the words from the passage that also appear in the query as the input of the CE_{CAT}. This suggests that CE_{CAT} re-rankers can be further improved by a better exact word matching, as the presence of query words in the document is one of the strongest signals for relevance in ranking [13, 14]. Moreover, obtaining improvement in effectiveness by interpolating the scores (score fusion [15]) of BM25 and CE_{CAT} is challenging: a linear combination of the two scores has shown to decrease effectiveness on the MSMARCO Passage collection compared to only using the CE_{CAT} re-ranker in the second stage retrieval [11].

To tackle this problem, in this work, we propose a method to enhance CE_{CAT} re-rankers illustrated in Fig. 1 by directly injecting the BM25 score as a string to the input of the Transformer. Figure 2 show our method for the injection of BM25 in the input of the CE re-ranker. We refer to our method as $CE_{BM25CAT}$. Our idea is inspired by the finding by Wallace et al. [16] that BERT models can capture numeracy. In this regard, we address the following research questions:

RQ1: What is the effectiveness of BM25 score injection in addition to the query and document text in the input of CE re-rankers?

This research question is designed to explore the integration of a commonly used lexical information retrieval model (BM25) with CE re-rankers, specifically examining how is the impact of the inclusion of the BM25 score on the CE re-rankers in terms of retrieval effectiveness. Addressing this question is crucial as it seeks to bridge the gap between lexical and CE re-rankers, potentially leading to improvements in the relevance and effectiveness. To answer this question we setup two experiments on three datasets: MSMARCO, TREC DL'19 and '20. First, since the BM25 score has no defined range, we investigate the effect of different representations of the BM25 score by applying various normalization methods. We also analyze the effect of converting the normalized scores of BM25 to integers. Second, we evaluate the best representation of BM25—based on our empricial study – on four cross-encoders: BERT-base, BERT-large [17], DistillBERT [18], and MiniLM [19], comparing $CE_{BM25CAT}$ to CE_{CAT} across different Transformer models with a smaller and larger number of parameters. Next, we compare our proposed approach to common interpolation approaches:

RQ2: What is the effectiveness of $CE_{BM25CAT}$ compared to common approaches for combining the final relevance scores of CE_{CAT} and BM25?

This research question is designed to empirically evaluate the effectiveness of the novel $CE_{BM25CAT}$ model against common methodologies that combine the relevance scores of CE_{CAT} and BM25. Addressing this question is essential as it highlights the improvements or drawbacks of adopting this new approach on the effectiveness of CE re-rankers. To analyze $CE_{BM25CAT}$ and CE_{CAT} in terms of exact matching compared to BM25 we address the following question:

RQ3: How effective can $CE_{BM25CAT}$ capture exact matching relevance compared to BM25 and CE_{CAT}?

This research question is designed to empirically assess the performance of the $CE_{BM25CAT}$ model in capturing exact matching relevance, which is essential to be analyzed compared to the BM25 where we inject the BM25 score into the input of CE re-rankers. By comparing $CE_{BM25CAT}$ with the traditional BM25 and the CE_{CAT} model, we aim to highlight the specific advantages or limitations of our proposed method. Next, we analyze what the optimal position for injecting the retriever score is, addressing the following question:

RQ4: What is the optimal position for injecting the retriever score: before the query, between query and document, or after the document?

This research question is designed to identify the most effective strategy for integrating the first-stage retriever score within the CE re-rankers, which is crucial to be identified in order to identify the most optimal design of our proposed method in terms of retrieval quality. We find out that injecting the relevance before the query is slightly more effective than injecting between query and document or after the document.

To investigate the generalizability of the proposed method, we assess if injecting other available relevance scores during re-ranking (i.e., a dense passage retriever's relevance score) also improves the effectiveness; we call this model CE_{DPRCAT}. To do so, we address the following question:

RQ5: What is the effectiveness of injecting the dense passage retriever (DPR) score into the input, and what is its optimal representation?

This research question is designed to explore the impact of injecting the score of dense passage retrievers (DPR) score into CE re-rankers. By assessing both the effectiveness and the optimal representation of the DPR score when injected into the retrieval input, this question addresses a crucial gap in understanding how CE re-rankers can be improved by utilizing of DPR score into their input representation. Identifying the optimal method for incorporating the DPR score will enable the development of more effective retrieval strategies, providing insights on how improving the effectiveness of search systems in real-world applications. Furthermore, to provide an explanation on the improvement of $CE_{BM25CAT}$, we perform a qualitative analysis of a case where CE_{CAT} fails to identify the relevant document that is found using $CE_{BM25CAT}$ with the help of the BM25 score.[1]

Moreover, previous studies have shown than using larger cross-encoder re-rankers as teachers for smaller cross-encoder re-rankers or dense retrievers leads to improvement of the effectiveness [6]. We analyze injecting the BM25 or DPR score into the input while training the student cross-encoders with using three larger cross-encoders as teachers to assess the effectiveness of injection together with knowledge distillation.

To the best of our knowledge, there is no prior work on the effectiveness of cross-encoder re-rankers by injecting a retrieval model's score into their input. Our main contributions in this work are seven-fold:

1. We introduce a novel strategy for optimizing the utilization of initial-stage retriever scores, including both BM25 and DPR scores, in cross-encoder re-rankers. Our approach demonstrates statistically significant improvements across all official metrics, confirmed through extensive experiments and detailed analysis.
2. Our investigation reveals superior performance of our method over traditional techniques that linearly interpolate the scores from BM25 and cross-encoders. We provide empirical evidence supporting the effectiveness of our non-linear integration strategy.
3. Through rigorous comparison, we establish the superior exact matching capabilities of our cross-encoder model, $CE_{BM25CAT}$, over the standard BM25, while highlighting the limitations of CE_{CAT} in similar scenarios.
4. We conduct a thorough analysis on the performance of CE_{CAT} and $CE_{BM25CAT}$ across various query types, demonstrating consistent superiority of $CE_{BM25CAT}$ over CE_{CAT} in handling diverse queries.
5. We prove the generalizability of our relevance score injection approach by incorporating DPR scores into the cross-encoder, CE_{DPRCAT}. This adaptation shows even greater effectiveness than $CE_{BM25CAT}$, further validating our method's robustness.
6. We find that injecting the relevance score into the student model could lead to improvement on the effectiveness in a teacher-student training setup.
7. We explore and analyze the impact of different positions for injecting retriever scores into the input of cross-encoder re-rankers. Our findings identify the most optimal position and provide insights into the positional effects of retriever score injection on performance of cross-encoder re-ranker.

After a discussion of related work in Sect. 2, we describe the retrieval models employed in section 3 and the specifics of our experiments and methods in Sect. 4. The results are examined and the research questions are addressed in Sect. 5. Finally, the conclusion is described in Sect. 6.

[1] In this work, we interchangeably use the words document and passage to refer to unit that should be retrieved.

2 Related work

2.1 Modifying the input of re-rankers

Boualili et al. [20, 21] propose a method for highlighting exact matching signals by marking the start and the end of each occurrence of the query terms by adding markers to the input. In addition, they modify original passages and expand each passage with a set of generated queries using Doc2query [22] to overcome the vocabulary mismatch problem. This strategy is different from ours in two aspects: (1) the type of information added to the input: they add four tokens as markers for each occurrence of query terms, adding a burden to the limited input length of 512 tokens for query and document together, while we only add the BM25 score. (2) The need for data augmentation: they need to train a Doc2query model to provide the exact matching signal for improving the BERT re-ranker while our strategy does not need any extra overhead in terms of data augmentation. A few recent, but less related examples are Al-Hajj et al. [23], who experiment with the use of different supervised signals into the input of the cross-encoder to emphasize target words in context and Li et al. [24], who insert boundary markers into the input between contiguous words for Chinese named entity recognition.

Additionally, there are various studies that show modifying cross-encoder re-ranker inputs by adding additional information represented as splitter tokens can improve their effectiveness [25–28]. BERT-FP [26] demonstrates that adding splitter tokens between each utterance of a dialogue can improve the effectiveness of BERT-based re-rankers for the passage response retrieval task. CLosER [25] proposes an expertise-aware post-training that modifies the input of BERT-based re-rankers with three different splitter tokens to differentiate the end of an utterance by the questioner, the end of an utterance by the professional responder with deep expertise, and the end of an utterance by the professional responder with shallow expertise. They show this boost the effectiveness of BERT-based re-rankers for conversational search in legal domain statistically significantly. Askari et al. [27] shows that injecting question tags alongside the question title and description in a modified fine-grained structured input can lead to improved effectiveness of cross-encoder re-rankers in answer retrieval in community question answering systems. Gretkowski et al. [28] shows that modifying the input of BERT-based classifiers by injecting commonsense concepts in addition to the question and solution can improve the effectiveness of BERT-based classifiers in terms of commonsense reasoning ability.

2.2 Numerical information in transformer models

The incorporation of numerical information into Transformer models encompasses a range of tasks [29] including basic arithmetic operations, numeration, magnitude comparison, arithmetic word problems, exact facts, measurement estimation, and numerical language modeling. Simple arithmetic tasks test models on basic operations such as addition and subtraction, often using synthetic datasets designed for both masked and causal language models [30, 31]. Numeration, or the task of decoding numeric strings into their corresponding values, has been explored with both static and contextualized embeddings [32, 33]. Magnitude comparison tasks assess the ability of models to determine the larger of two or more numbers [32]. Arithmetic word problems and exact facts reasoning require models to apply numerical knowledge in more complex textual scenarios,with several datasets challenging models with such tasks [34–39]. Measurement estimation further explores models' capabilities in approximating and comparing quantities,using diverse benchmarks [40–44]. Numerical language modeling, an extension of traditional language modeling,involves predicting numeric values in text, evaluated through regression metrics [30]. Downstream applications of these numeracy capabilities are extensive, ranging from sarcasm detection in tweets to enhancing performance on numeric-heavy QA tasks [30, 45, 46], showcasing both the utility and versatility of Transformer models in handling numerical data.

Wallace et al. [16] analyze the ability of BERT models to work with numbers and come to the conclusion that the models capture numeracy and are able to do numerical reasoning; however the models appeared to struggle with interpreting floats. Moreover, Zhang et al. [47] show that BERT models capture a significant amount of information about numerical scale except for general common-sense reasoning. There are various studies that are inspired by the fact that Transformer models can correctly process numbers [28, 46, 48–51]. Gu et al. [52] incorporate text, categorical and numerical data as different modalities with Transformers using a combining module accross different

classification tasks. They discover that adding tabular features increases the effectiveness while using only text is insufficient and results in the worst performance. Recently, [53] has shown the impact of injection credibility score as statements into the input of cross-encoder re-rankers for guiding re-rankers to retrieve not only relevant but also credible documents on top of relevant but less credible documents.

2.3 Methods for combining rankers

Linearly interpolating different rankers' scores has been studied extensively in the literature [11, 15, 54–56]. In information retrieval (IR), various methodologies have been developed for combining search results from multiple rankers. These strategies can be categorized into score-based, rank-based, probabilistic, and voting-based methods, each addressing different aspects of the fusion challenge [57]. Score-based methods integrate the relevance scores provided by individual search engines to formulate the final relevance score [58–60]. Rank-based methods [61–63], in contrast, do not rely on explicit relevance scores but instead utilize the ordinal positions of documents across different search results. This method is particularly valuable when dealing with data from web search aggregators, such as Kayak and Skyscanner, where individual relevance scores might not be accessible. Rank-based strategies focus solely on the synthesis of ranking positions, streamlining the fusion process under constraints of limited data. Probabilistic methods [64–67] introduce a statistical approach by estimating the probability distribution of document relevance across ranking positions for each search engine. These methods necessitate a training phase to accurately model these probabilities, thus adding a layer of complexity but potentially increasing the accuracy of the fused rankings by incorporating probabilistic inference. Voting-based methods [64, 68] adapt traditional voting procedures to metasearch. Methods like the Borda Count and the Condorcet method are used to amalgamate the preferences of multiple search engines, treating each engine as an "expert" voter. This approach is often seen as an extension of rank-based methods, as it primarily relies on the ranks rather than scores.

In this paper, we investigate multiple score-based linear and non-linear interpolation ensemble methods to analyze the performance of them for combining BM25 and CE_{CAT} scores in comparison to $CE_{BM25CAT}$. For the sake of a fair analysis, we do not compare $CE_{BM25CAT}$ with a Learning-to-rank approach that is trained on 87 features by Zhang et al. [69]. The use of ensemble methods brings additional overhead in terms of efficiency because it adds one more extra step to the re-ranking pipeline. It is noteworthy to mention that in this paper, we concentrate on analyzing the improvement by combining the first-stage retriever and a BERT-based re-ranker: BM25 and CE_{CAT} respectively. However, we are aware that combining scores of BM25 and Dense Retrievers that both are first-stage retrievers has also shown improvements [70–72] that are outside the scope of our study. In particular, CLEAR [10] proposes an approach to train the dense retrievers to encode semantics that BM25 fails to capture for first stage retrieval. However, in this study, our aim is to improve re-ranking in the second stage of two-stage retrieval setting.

2.4 Knowledge distillation

Knowledge distillation has become an important approach in information retrieval [6, 73]. In machine learning, knowledge distillation refers to the process of transferring knowledge from a more capable model (called the teacher) to a less capable model (called the student) [74]. Following this convention, the goal in information retrieval is to train a small ranker (the student) with one or multiple more larger ranker models (the teachers) on a given dataset.

Knowledge distillation methods can be categorized into methods that employ multiple teachers of one type of model, called teacher ensembles, or methods that use different types of models as teachers [75]. In this paper, where we experiment with knowledge distillation, we use a highly effective method called Marginal Mean Square Error (Margin-MSE) [6] which uses the ensemble of three relatively large BERT-based cross-encoder re-rankers as teachers and a smaller BERT-based cross-encoder re-ranker as the student. Margin-MSE utilizes the margin between the scores of relevant and non-relevant passages as distilled knowledge.

3 Methods

In the following, we first describe the first-stage retrievers and re-rankers that are used and then describe the proposed method in detail.

3.1 First stage rankers

We experiment with two widely used first-stage rankers namely BM25 [76] and dense passage retrievers [77].

3.1.1 BM25

Lexical retrievers estimate the relevance of a document to a query based on word overlap [76]. Many lexical methods, including vector space models, Okapi BM25, and query likelihood, have been developed in previous decades. We use BM25 because of its popularity as first-stage ranker in current systems. Based on the statistics of the words that overlap between the query and the document, BM25 calculates a score for the pair:

$$\begin{aligned} s_{lex}(q,d) &= BM25(q,d) \\ &= \sum_{t \in q \cap d} rsj_t . \frac{tf_{t,d}}{tf_{t,d} + k_1 \left\{ (1-b) + b \frac{|d|}{l} \right\}} \end{aligned} \tag{1}$$

where t is a term, $tf_{t,d}$ is the frequency of t in document d, rsj_t is the Robertson-Spärck Jones weight [2] of t, and l is the average document length. k_1 and b are parameters [78, 79].

3.1.2 DPR

Dense passage retrieval (DPR) models [77] provide an efficient BERT-based first-stage retriever that estimates relevance beyond word overlap by matching query and document text in a continuous representation space. In contrast to cross-encoder re-rankers that process both the query and document at the same time to estimate relevance, which makes the process computationally heavy, the DPR model processes this in two phases, significantly improving efficiency and making DPR models an efficient alternative first-stage retriever. The representations of the collection's passages are pre-computed in an offline setup, where a single CLS vector represents the contextualized representation of a passage (**p**). Given a query, DPR models represent the query as a single CLS vector (**q**) and estimate the relevance of a document to a query based on the dot product of their corresponding CLS vectors. This is also called BERT_{DOT} [6], formally defined as:

$$BERT_{DOT}(q_{1:m}, p_{1:n}) = \mathbf{q} \cdot \mathbf{p} \tag{2}$$

3.2 CE_{CAT}: cross-encoder re-rankers without BM25 injection

Concatenating query and passage input sequences is the typical method for using cross-encoder (e.g., BERT) architectures with pre-trained Transformer models in a re-ranking setup [4, 6, 80, 81]. This basic design is referred to as CE_{CAT} and shown in Fig. 1. The query $q_{1:m}$ and passage $p_{1:n}$ sequences are concatenated with the [*SEP*] token, and the [*CLS*] token representation computed by CE is scored with a single linear layer W_s in the CE_{CAT} ranking model:

$$CE_{CAT}(q_{1:m}, p_{1:n}) = CE([CLS]\, q\, [SEP]\, p\, [SEP]) * W_s \tag{3}$$

We use CE_{CAT} as our baseline re-ranker architecture. We evaluate different cross-encoder models in our experiments and all of them follow the above design.

3.3 $\text{CE}_{\text{BM25CAT}}$: cross-encoder re-rankers with BM25 injection

To study the effectiveness of injecting the BM25 score into the input, we modify the input of the basic input format as follows and call it $\text{CE}_{\text{BM25CAT}}$:

$$\begin{aligned} &CE_{BM25CAT}(q_{1:m}, p_{1:n}) \\ &\quad = CE([CLS]\, q\, [SEP]\, BM25\, [SEP]\, p\, [SEP]) * W_s \end{aligned} \tag{4}$$

where BM25 represent the relevance score produced by BM25 between query and passage.

We study different representations of BM25 to find the optimal approach for injecting BM25 into the cross-encoders. The reasons are: (1) BM25 scores do not have an upper bound and should be normalized for having an interpretable score given a query and passage; (2) BERT-based models can process integers better than floating point numbers [16] so we analyze if converting the normalized score to an integer is more effective than injecting the floating point score. For normalizing BM25 scores, we compare three different normalization methods: Min-Max, Standardization[2] (Z-score), and Sum:

$$Min\text{-}Max(s_{BM25}) = \frac{s_{BM25} - s_{min}}{s_{max} - s_{min}} \tag{5}$$

$$Standard(s_{BM25}) = \frac{s_{BM25} - \mu(S)}{\sigma(S)} \tag{6}$$

$$Sum(s_{BM25}) = \frac{s_{BM25}}{sum(S)} \tag{7}$$

where s_{BM25} is the original score, and s_{max} and s_{min} are the maximum and minimum scores respectively, in the ranked list. $Sum(S)$, $\mu(S)$, and $\sigma(S)$ refer to sum, average and standard deviation over the scores of all passages retrieved for a query. The anticipated effect of the Sum normalizer is that the sum of the scores of all passages in the ranked list will be 1; thus, if the top-n passages receive much higher scores than the rest, their normalized scores will have a larger difference with the rest of passages' scores in the ranked list; this distance could give a good signal to $CE_{BM25CAT}$. We experiment with Min-Max and Standardization in a local and a global setting. In the local setting, we get the minimum or maximum (for Min-Max) and mean and standard deviation (for Standard) from the ranked list of scores per query. In the global setting, we use $\{0, 50, 42, 6\}$ as {minimum, maximum, mean, standard deviation} as they have been empirically suggested in prior work to be used as default values across different queries to globally normalize BM25 scores [82]. In our data, the {minimum, maximum, mean, standard deviation} values are $\{0, 98, 7, 5\}$ across all queries. Because of the differences between the recommended defaults and the statistics of our collections, we explore other global values for Min-Max, using 25, 50, 75, 100 as maximum and 0 as minimum. However, we got the best result using default values of [82]. To convert the float numbers to integers we multiply the normalized score to 100 and discard decimals. Finally, we store the number as a string.

3.4 Linear interpolation ensembles of BM25 and CE_{CAT}

We compare our approach to common ensemble methods [11, 83] for interpolating BM25 and BERT re-rankers. We combine the scores linearly using the following methods: (1) Sum: compute sum over BM25 and CE_{CAT} scores, (2) Max: select maximum between BM25 and CE_{CAT} scores, and (3) Weighted-Sum:

$$s_i = \alpha \,.\, s_{BM25} + (1 - \alpha) \,.\, s_{CE_{CAT}} \tag{8}$$

where s_i is the weighted sum produced by the interpolation, s_{BM25} is the normalized BM25 score, $s_{CE_{CAT}}$ is the CE_{CAT} score, and $\alpha \in [0..1]$ is a weight that indicates the relative importance. Since CE_{CAT} score $\in [0, 1]$, we also normalize BM25 score using Min-Max normalization. Furthermore, we train ensemble models that take s_{BM25} and $s_{CE_{CAT}}$ as features. We experiment with three different classifiers for this purpose: SVM with a linear kernel, SVM with an RBFkernel, Naive Bayes, and Multi Layer Perceptron (MLP) as a non-linear method and report the best classifier performance in Sect. 5.3.

3.5 CE_{DPRCAT}: cross-encoder re-rankers with DPR injection

We inject the DPR score into the input of the cross-encoders, similarly to $CE_{BM25CAT}$. We compare different representations of DPR scores to find the optimal approach for injecting DPR into the cross-encoders. This is because the relevance score computed by calculating the dot product is not normalized to a fixed range. Therefore, in addition to injecting the original

[2] In this context, "Standard" is used as an alternative name for z-score normalization, which relies on the standard deviation.

score, we investigate Min-Max normalization in global and local setting with float and integer representations. We use a pre-trained BERT-Base$_{DOT}$ as our DPR model which is a dense retrieval model trained with knowledge distillation [6].[3]

3.6 Knowledge distillation with score injection

Previous studies have shown than using larger cross-encoder re-rankers as teachers for smaller cross-encoder re-rankers or dense retrievers leads to improvement of the effectiveness. We analyze injecting the BM25 and DPR score in the input while training the student cross-encoders with using three larger cross-encoders as teachers. To do so, following [6], we train MiniLM to predict the prediction of the ensemble score of 3 large teachers models, BERT-Base$_{CAT}$, BERT-Large-WM$_{CAT}$, and ALBERT-Large$_{CAT}$, per query–document sample. We use Mean Square Error (MSE) as loss function and the scores of the teacher models published by Hofstätter et al. [6].[4] This gives MiniLM performances comparable to large models, while being 18 times faster. Our motivation for doing so is analyzing if even if a such scenario, injecting the BM25 or DPR score can lead to improvement of effectiveness.

4 Experimental design

4.1 Dataset and metrics

We conduct our experiments on the MSMARCO-passage collection [84] and the two TREC Deep Learning tracks (TREC-DL'19 and TREC-DL'20) [85, 86]. The MSMARCO-passage dataset contains about 8.8 million passages (average length: 73.1 words) and about 1 million natural language queries (average length: 7.5 words) and has been extensively used to train deep language models for ranking because of the large number of queries. Following prior work on MSMARCO [11, 87–90], we use the dev set ($\sim 7k$ queries) for our empirical evaluation. *MAP*@1000 and *nDCG*@10 are calculated in addition to the official evaluation metric *MRR*@10. The passage corpus of MSMARCO is shared with TREC DL'19 and DL'20 collections with 43 and 54 queries respectively. We evaluate our experiments on these collections using *nDCG*@10 and *MAP*@1000, as is standard practice in TREC DL [85, 86] to make our results comparable to previously published and upcoming research. We cap the query length at 30 tokens and the passage length at 200 tokens following prior work [6].

4.2 Training configuration and model parameters

We use the Huggingface library [91], Cross-encoder package of Sentence-transformers library [92], and PyTorch [93] for the cross-encoder re-ranking training and inference. We fine-tune all of the cross-encoder re-rankers in a similar configuration to ensure the fairness and reliability of our comparison. We train a distinct cross-encoder re-ranker for each query set in the evaluation. For each TREC DL collection, we use the other TREC DL query set as the validation set, and we select both TREC DL ('19 and '20) query sets as the validation set to train CEs for the MSMARCO DEV Passage collection. Please note that the train/validation/test data are the same within the compared CE$_{CAT}$, CE$_{BM25CAT}$, and CE$_{DPRCAT}$ models to ensure the only difference during fine-tuning and evluation a cross-encoder re-ranker is the presence or exclusion of score injection in the input. For the dense passage retriever model, we use an already existing state-of-the-art dense passage retrieval model that is trained on the MSMarco training set.[5]

For injecting the BM25 or DPR score as text, we pass the score in string format into the BERT tokenizer in a similar way to passing query and document. Please note that the integer numbers are already included in the BERT tokenizer's vocabulary, allowing for appropriate tokenization. Following prior work [6] we use the Adam [94] optimizer with a learning rate of $7 * 10^{-6}$ for all cross-encoder layers, regardless of the number of layers trained. We employ early stopping, based on the nDCG@10 value of the validation set. We use a training batch size of 32. For all cross-Encoder re-rankers, we use Cross-Entropy loss [95]. For the lexical retrieval with BM25 we employ the tuned parameters from the Anserini documentation [78, 79].[6]

[3] https://huggingface.co/sebastian-hofstaetter/distilbert-dot.

[4] https://github.com/sebastian-hofstaetter/neural-ranking-kd.

[5] https://huggingface.co/sentence-transformers/msmarco-distilbert-base-tas-b

[6] The code is available on https://github.com/arian-askari/injecting_bm25_score_bert..

Table 1 Effectiveness results

	Normalization	Local/Global	Float/Integer	MSMARCO DEV		
				nDCG@10	MAP	MRR@10
(*a*)	$\text{MiniLM}_{\text{CAT}}$ (without injecting BM25 score)			0.419	0.363	0.360
(*b*)	Original Score	–	–	0.420	0.364	0.362
(*c*)	Min-Max	Local	Float	0.411	0.359	0.354
(*d*)	Min-Max	Local	Integer	0.414	0.361	0.355
(*e*)	Min-Max	Global	Float	0.422	0.365	0.363
(*f*)	Min-Max	Global	Integer	**0.424**†	**0.368**†	**0.367**†
(*g*)	Standard	Local	Float	0.407	0.355	0.352
(*h*)	Standard	Local	Integer	0.410	0.358	0.354
(*i*)	Standard	Global	Float	0.420	0.363	0.361
(*j*)	Standard	Global	Integer	0.421	0.365	0.363
(*k*)	Sum	–	Float	0.402	0.349	0.338
(*l*)	Sum	–	Integer	0.405	0.350	0.342

Bold face indicates best performance

Lines *b*–*n* refer to the $\text{MiniLM}_{\text{BM25CAT}}$ re-ranker using different representations of the BM25 score as text. The position of score injection is in the middle, between the query and the candidate document

Significance is shown with † for the best result (row *f*) compared to $\text{MiniLM}_{\text{CAT}}$ (row *a*). Statistical significance was measured with a paired t-test ($p < 0.05$) with Bonferroni correction for multiple testing

4.3 Knowledge distillation

For knowledge distillation (KD), we use mean square error loss followed by Hofstätter et al. [6]. For the teachers' relevance scores, we use the publicly available scores for pairs of queries and documents from the MSMARCO training set released by Hofstätter et al. [6]. For injection in the KD setup, we compute and inject either BM25 or DPR score.

5 Results

5.1 Choice of BM25 score representation

As introduced in Sect. 3.3, we compare different representations of the BM25 score in Table 1 for injection into $\text{CE}_{\text{BM25CAT}}$. We chose MiniLM [19] for this study as it has shown competitive results in comparison to BERT-based models while it is 3 times smaller and 6 times faster.[7] Our first interesting observation is that injecting the original float score rounded down to 2 decimal points (row *b*) of BM25 into the input seems to slightly improve the effectiveness of re-ranker. We assume this is due to the fact that the average query and passage length is relatively small in the MSMARCO Passage collection, which prevents from getting high numbers—with low interpretability for BERT—as BM25 score. Second, we find that the normalized BM25 score with Min-Max in the global normalization setting converted to integer (row *f*) is the most significant effective[8]. representation for injecting BM25.

The global normalization setting gives better results for both Min-Max (rows *e*, *f*) and Standardization (rows *i*, *j*) than local normalization (rows *c*, *d* and *g*, *h*).[9] The reason is probably that in the global setting a candidate document obtains a high normalized score (close to 1 in the floating point representation) if its original score is close to default maximum (for Min-Max normalization) so the normalized score could be more interpretable across different queries. On the other hand, in the local setting, the passages ranked at position 1 always receive 1 as normalized score with Min-Max even if its original score is not high and it does not have a big difference with the last passage in the ranked list.

[7] https://huggingface.co/microsoft/MiniLM-L12-H384-uncased

[8] Although the evaluation metrics are not in an interval scale, Craswell et al. [96] show that they are mostly reliable in practice on MSMARCO for statistical testing

[9] The range of normalized integer scores using the best normalizer (row *f*) are from 0 to 196 as the maximum BM25 score in the collection is 98.

Table 2 Effectiveness results

Model	TREC DL 20		TREC DL 19		MSMARCO DEV		
	nDCG@10	MAP	nDCG@10	MAP	nDCG@10	MAP	MRR@10
BM25	0.480	0.286	0.506	0.377	0.234	0.195	0.187
Re-rankers							
BERT- Base_{CAT}	0.689	0.447	0.713	0.441	0.399	0.346	0.342
BERT- $\text{Base}_{\text{BM25CAT}}$	0.705†	0.475†	0.723†	0.453†	0.422†	0.367†	0.364†
BERT-$\text{Large}_{\text{CAT}}$	0.695	0.464	0.714	0.467	0.401	0.344	0.360
BERT-$\text{Large}_{\text{BM25CAT}}$	**0.728**†	**0.482**†	**0.731**†	**0.477**†	**0.424**†	0.367†	**0.369**†
$\text{DistilBERT}_{\text{CAT}}$	0.670	0.442	0.679	0.440	0.383	0.310	0.325
$\text{DistilBERT}_{\text{BM25CAT}}$	0.682†	0.456†	0.699†	0.451†	0.390†	0.323†	0.339†
$\text{MiniLM}_{\text{CAT}}$	0.681	0.448	0.704	0.452	0.419	0.363	0.360
$\text{MiniLM}_{\text{BM25CAT}}$	0.710†	0.473†	0.711†	0.463†	**0.424**†	**0.368**†	0.367†

Bold face indicates best performance

Fine-tuned cross-encoders are used for re-ranking over BM25 first stage retrieval with a re-ranking depth of 1000. Statistical significance was measured with a paired t-test ($p < 0.05$) with Bonferroni correction for multiple testing. The position of score injection is in the middle, between the query and the candidate document

† indicates a statistically significant improvement of a cross-encoder with BM25 score injection as text into the input (Cross-$\text{encoder}_{\text{BM25CAT}}$) over the same cross-encoder without BM25 score injection (Cross-$\text{encoder}_{\text{CAT}}$)

Moreover, converting the normalized float score to integers gives better results for both Min-Max (rows *d*, *f*) and Standardization (rows *h*, *j*) than the float representation (rows *c*, *e* and *g*, *i*). We find that Min-Max normalization is a better representation for injecting BM25 than Standardization, which could be due to the fact that in Min-Max the normalized score could not be negative, and, as a result, interpreting the injected score is easier for $\text{CE}_{\text{BM25CAT}}$. We find that the Sum normalizer (rows *k* and *l*) decreases effectiveness. Apparently, our expectation that Sum would help distinguish between the top-*n* passages and the remaining passages in the ranked list (see Sect. 5.1) is not true.

5.2 Impact of BM25 injection for various cross-encoders (RQ1)

Table 2 shows that injecting the BM25 score – using the best normalizer which is Min-Max in the global normalization setting converted to integer – into all four cross-encoders improves their effectiveness in all of the metrics compared to using them without injecting BM25. This shows that injecting the BM25 score into the input as a small modification to the current re-ranking pipeline improves the re-ranking effectiveness. This is without any additional computational burden as we train CE_{CAT} and $\text{CE}_{\text{BM25CAT}}$ in a completely equal setting in terms of number of epochs, batch size, etc. We receive the highest result by $\text{BERT-Large}_{\text{BM25CAT}}$ for cross-encoder with BM25 injection, which could be due to the higher number of parameters of the model. We find that the results of MiniLM are similar to those for BERT-Base on MSMARCO-DEV while the former is more efficient.

5.3 Comparing BM25 Injection with Ensemble Methods (RQ2)

Table 3 shows that while injecting BM25 leads to improvement, regular ensemble methods and Naive Bayes classifier fail to do so; combining the scores of BM25 and BERT_{CAT} in a linear and non-linear (MLP) interpolation ensemble setting even leads to lower effectiveness than using the cross-encoder as sole re-ranker. Therefore, our strategy is a better solution than linear interpolation. We only report results for Naive Bayes—having BM25 and BERT_{CAT} score as features—as it had the highest effectiveness of the four estimators. Still, the effectiveness is much lower than $\text{BERT}_{\text{BM25CAT}}$ and also lower than a simple Weighted-Sum. Weighted-Sum (tuned) in Table 3 is tuned on the validation set, for which $\alpha = 0.1$ was found to be optimal. We analyze the effect of different α values in a weighted linear interpolation (Weighted-Sum) to draw a more complete picture on the impact of combining scores on the DEV set. Figure 3 shows that by increasing the weight of BM25, the effectiveness decreases. The figure also shows that the tuned alpha which was found on the validation set in Table 3 is not the most optimal possible alpha value for the DEV

Table 3 The effectiveness of injecting BM25 score into the input (Bert-Base$_{BM25CAT}$) compared to interpolation performance of BM25 and Bert-Base$_{CAT}$ using common ensemble methods

Model	Ensemble	MSMARCO DEV		
		nDCG@10	MAP	MRR@10
BM25	–	0.234	0.195	0.187
BERT-Base$_{CAT}$	–	0.399	0.346	0.342
BM25 and BERT-Base$_{CAT}$	Sum	0.270	0.225	0.218
BM25 and BERT-Base$_{CAT}$	Max	0.237	0.197	0.190
BM25 and BERT-Base$_{CAT}$	Weighted-Sum (tuned)	0.353	0.295	0.290
BM25 and BERT-Base$_{CAT}$	Naive Bayes	0.314	0.260	0.254
BERT-Base$_{BM25CAT}$	BM25 Score Injection	**0.422**	**0.367**	**0.364**

Bold face indicates best performance

The position of score injection is in the middle, between the query and the candidate document

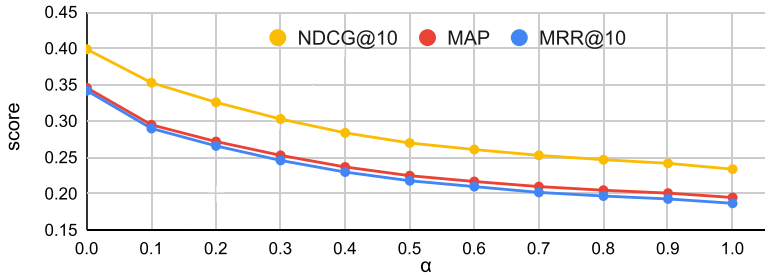

Fig. 3 Effectiveness on MSMARCO DEV with varying the interpolation weight of BM25 and BERT-Base$_{CAT}$ scores. $\alpha = 0$ means only BERT$_{CAT}$ scores are used

set. The highest effectiveness for $\alpha = 0.0$ in Fig. 3 confirms we should not combine the scores by current interpolation methods and only using scores of Bert-Base$_{CAT}$ is better, at least for the MSMARCO passage collection.

5.4 Exact matching relevance results (RQ3)

To conduct exact matching analysis, we replace the passage words that do not appear in the query with the [*MASK*] token, leaving the model only with a skeleton of the original passage and force it to rely on the exact word matches between query and passage [12]. We do not train models on this input but use our models that were fine-tuned on the original data. Table 4 shows that BERT-Base$_{BM25CAT}$ performs better than both BM25 and BERT-Base$_{CAT}$ in the exact matching setting on all metrics. Moreover, we found that the percentage of relevant passages ranked in top-10 that are common between BM25 and BERT$_{BM25CAT}$ is 40%, which is higher than the percentage of relevant passages between BM25 and BERT$_{CAT}$ (37%). Therefore, the higher effectiveness of BERT$_{BM25CAT}$ in exact matching setting could be at least partly because it mimics BM25 more than BERT$_{CAT}$. In comparison, this percentage is 57 between BERT$_{BM25CAT}$ and BERT$_{CAT}$.

Table 4 Comparing exact matching effectiveness of BERT-Base$_{BM25CAT}$ and BERT-Base$_{CAT}$ by keeping only the query words in each passage for re-ranking

Model	Input	MSMARCO DEV		
		nDCG@10	MAP	MRR@10
BM25	Full text	0.234	0.195	0.187
BERT-Base$_{CAT}$	Only query words	0.218 (↓1.6)	0.186 (↓0.9)	0.180 (↓0.7)
BERT-Base$_{BM25CAT}$	Only query words	**0.243** (↑0.9)	**0.209** (↑1.4)	**0.202** (↑1.5)

Bold face indicates best performance

The increase and decrease of effectiveness compared to BM25 is indicated with ↑ and ↓. The position of score injection is in the middle, between the query and the candidate document

Table 5 Analyzing the position of the BM25 score injection

Position of injection	TREC DL 20		TREC DL 19		MSMARCO DEV		
	nDCG@10	MAP	nDCG@10	MAP	nDCG@10	MAP	MRR@10
–	0.681	0.448	0.704	0.452	0.419	0.363	0.360
Between q and d	0.710	0.473	0.711	0.463	0.424	0.368	0.367
After d	0.712	0.475	0.713	0.466	0.425	0.369	0.368
Before q	0.717	0.483	0.718	0.469	0.428	0.373	0.374

q and d refer to query and document

5.5 Impact of position of injection (RQ4)

To assess whether injection position has an impact on the effectiveness of $CE_{BM25CAT}$, we try three different positions for injecting the BM25 relevance score into the input of $CE_{BM25CAT}$:

- Between query and candidate document, in which the position of the injection is influenced by the query length: '[CLS] query [SEP] BM25 [SEP] document [SEP]'
- Before the query, in which the position of the injection is independent from query or document length: '[CLS] BM25 [SEP] query [SEP] document [SEP]'
- After candidate document, in which the position of injection is dependent on the query and document length: '[CLS] query [SEP] document [SEP] BM25 [SEP]'

The motivation behind this analysis comes from the fact that injecting the BM25 score in the middle, between query and candidate document, gives a different position each time to the BM25 score, while there is not any actual meaningful difference based on the position of the BM25 score. As a result, using a fix position for injection might lead to the improvement. Table 5 confirms this analysis and shows that injecting BM25 before the query achieves the highest effectiveness. Since the gain of modifying the position of injection is not significant, we do not re-do the previous experiments and keep the position of injection in previous experiments in the middle as explained in Sect. 3.

5.6 Effectiveness of injecting DPR into CE_{DPRCAT} (RQ5)

5.6.1 Choice of DPR score representation

As introduced in Sect. 3.5, we compare different representations of the DPR score in Table 6 for injection into CE_{DPRCAT} by using Min-Max as normalization strategy. We chose MiniLM [19] for this study similar to the previous analysis for

Table 6 Effectiveness results

	Normalization	Local/Global	Float/Integer	MSMARCO DEV		
				nDCG@10	MAP	MRR@10
(*a*)	$MiniLM_{CAT}$ (without injecting DPR score)			0.404	0.347	0.345
(*b*)	Original score	–	–	0.405	0.347	0.346
(*c*)	Min–Max	Local	Float	0.395	0.341	0.340
(*d*)	Min–Max	Local	Integer	0.399	0.343	0.341
(*e*)	Min–Max	Global	Float	0.442	0.387	386
(*f*)	Min–Max	Global	Integer	**0.450**†	**0.395**†	**0.394**†

Bold face indicates best performance

Lines *b-n* refer to the $MiniLM_{DPRCAT}$ re-ranker using different representations of the DPRCAT score as text. The initial ranker is DPR and re-ranking depth is 1000

Significance is shown with † for the best result (row *f*) compared to $MiniLM_{CAT}$ (row *a*). Statistical significance was measured with a paired t-test ($p < 0.05$) with Bonferroni correction for multiple testing

Table 7 Effectiveness results of injecting DPR and BM25 scores using DPR or BM25 as initial ranker

Ranker	Re-ranker	TREC'20		TREC'19		MSMARCO DEV		
		nDCG@10	MAP@1k	nDCG@10	MAP@1k	nDCG@10	MAP@1k	MRR@10
BM25	–	0.480	0.286	0.506	0.377	0.234	0.195	0.187
DPR	–	0.685	0.475	0.721	0.476	0.408	0.352	0.347
BM25	$\text{MiniLM}_{\text{CAT}}$	0.681	0.448	0.704	0.452	0.419	0.363	0.360
	$\text{MiniLM}_{\text{BM25CAT}}$	0.710†	0.473†	0.711†	0.463†	0.424†	0.368†	0.367†
	$\text{MiniLM}_{\text{DPRCAT}}$	0.712*	0.485*	0.713*	0.467*	0.429*	0.372*	0.371*
DPR	$\text{MiniLM}_{\text{CAT}}$	0.678	0.470	0.702	0.472	0.404	0.347	0.345
	$\text{MiniLM}_{\text{BM25CAT}}$	0.741†	0.495†	0.728†	0.489†	0.441†	0.389†	0.388†
	$\text{MiniLM}_{\text{DPRCAT}}$	**0.746***	**0.498***	**0.730***	**0.490***	**0.450***	**0.395***	**0.394***

Bold face indicates best performance

Statistical significance was measured with a paired t-test ($p < 0.05$) with Bonferroni correction for multiple testing. The position of score injection is in the middle, between the query and the candidate document

† Indicates a statistically significant improvement of a cross-encoder with DPR score injection as text into the input (Cross-encoder$_{\text{BM25CAT}}$) over the same cross-encoder without BM25 score injection (Cross-encoder$_{\text{CAT}}$)

* Indicates a statistically significant improvement of a cross-encoder with DPR score injection as text into the input (Cross-encoder$_{\text{DPRCAT}}$) over the same cross-encoder without DPR score injection (Cross-encoder$_{\text{CAT}}$)

$\text{CE}_{\text{BM25CAT}}$. We observe a similar pattern to $\text{CE}_{\text{BM25CAT}}$ for injecting DPR into $\text{CE}_{\text{DPRCAT}}$: First, injecting the original float score rounded down to 2 decimal points DPR into the input seems to slightly improve the effectiveness of re-ranker. Second, we find that the normalized score with Min-Max in the global normalization setting converted to integer is the most effective representation for injecting DPR. We found 118 as global maximum value and 89 as global minimum value in the DPR scores for the MS MARCO training set and used them as global values for Min-Max. The position of score injection is in the middle, between the query and the candidate document.

5.6.2 Finding the most effective two-stage setting

To determine the optimal configuration for $\text{CE}_{\text{BM25CAT}}$ and $\text{CE}_{\text{DPRCAT}}$, we employed two initial rankers: BM25 and DPR, followed by two re-rankers: $\text{CE}_{\text{BM25CAT}}$ and $\text{CE}_{\text{DPRCAT}}$. Table 7 presents our findings, which reveals that utilizing DPR as the initial ranker in combination with $\text{CE}_{\text{DPRCAT}}$ as the re-ranker yields the highest effectiveness. This combination slightly outperforms the use of $\text{CE}_{\text{BM25CAT}}$ as the re-ranker in the same setup.

Additionally, we observed that re-ranking *without* injecting, using the effective DPR model as the intial ranker, does not consistently result in improved effectiveness over only the first-stage retrieval. Specifically, when considering nDCG@10 for TREC'19, DPR achieves a score of 0.721, while $\text{MiniLM}_{\text{CAT}}$ scores 0.717. This is while, employing either $\text{MiniLM}_{\text{BM25CAT}}$ or $\text{MiniLM}_{\text{DPRCAT}}$ as the re-ranker consistently leads to significant improvements over DPR.

Furthermore, we found out that $\text{MiniLM}_{\text{BM25CAT}}$ is less effective than $\text{MiniLM}_{\text{DPRCAT}}$ for re-ranking over BM25 as the initial ranker. This shows even if BM25 is used as first-stage retriver, there is still a benefit in terms of effectiveness by using $\text{MiniLM}_{\text{DPRCAT}}$ as the re-ranker. The position of score injection is in the middle, between the query and the candidate document.

5.6.3 Impact of DPR injection for various cross-encoders

Table 8 shows that injecting the DPR score—using the best normalizer which is Min–Max in the global normalization setting converted to integer—into all four cross-encoders improves their effectiveness in all of the metrics compared to using them without injecting DPR. This shows that injecting the DPR score into the input as a small modification to the current re-ranking pipeline improves the re-ranking effectiveness. This is without any additional computational burden as we train CE_{CAT} and $\text{CE}_{\text{DPRCAT}}$ in a completely equal setting in terms of number of epochs, batch size, etc. We receive the highest result by $\text{BERT-Large}_{\text{DPRCAT}}$ with DPR injection, which is likely be due to the large number of parameters of the model.

Table 8 Effectiveness results

Model	TREC'20		TREC'19		MSMARCO DEV		
	nDCG@10	MAP	nDCG@10	MAP	nDCG@10	MAP	MRR@10
BM25	0.480	0.286	0.506	0.377	0.234	0.195	0.187
DPR	0.685	0.475	0.721	0.476	0.408	0.352	0.347
Re-rankers							
BERT-BaseCAT	0.683	0.471	0.707	0.473	0.406	0.349	0.347
BERT- $\text{Base}_{\text{DPRCAT}}$	0.750†	0.515†	0.734†	0.492†	0.452†	0.396†	0.397†
BERT-LargeCAT	0.691	0.482	0.709	0.476	0.410	0.353	0.350
BERT-$\text{Large}_{\text{DPRCAT}}$	0.757†	0.527†	0.741†	0.505†	0.466†	0.401†	0.408†
DistilBERTCAT	0.672	0.467	0.696	0.468	0.397	0.339	0.338
$\text{DistilBERT}_{\text{DPRCAT}}$	0.721†	0.488†	0.726†	0.481†	0.493†	0.383†	0.388†
MiniLMCAT	0.678	0.470	0.702	0.472	0.404	0.347	0.345
$\text{MiniLM}_{\text{DPRCAT}}$	0.746†	0.498†	0.730†	0.490†	0.450†	0.395†	0.394†

Fine-tuned cross-encoders are used for re-ranking over DPR first stage retrieval with a re-ranking depth of 1000

Statistical significance was measured with a paired t-test ($p < 0.05$) with Bonferroni correction for multiple testing. The position of score injection is in the middle, between the query and the candidate document

† Indicates a statistically significant improvement of a cross-encoder with DPR score injection as text into the input (Cross-$\text{encoder}_{\text{DPRCAT}}$) over the same cross-encoder without DPR score injection (Cross-$\text{encoder}_{\text{CAT}}$)

Table 9 Effectiveness results of injecting DPR and BM25 scores using DPR or BM25 as initial ranker in a knowledge distillation training setup

Ranker	Re-ranker	TREC DL 20		TREC DL 19		MSMARCO DEV		
		NDCG@10	MAP@1k	NDCG@10	MAP@1k	NDCG@10	MAP@1k	MRR@10
BM25	–	0.480	0.286	0.506	0.377	0.234	0.195	0.187
DPR	–	0.685	0.475	0.721	0.476	0.408	0.352	0.347
BM25	$\text{MiniLM}_{\text{CAT}}$	0.732	0.477	0.715	0.483	0.412	0.352	0.350
	$\text{MiniLM}_{\text{BM25CAT}}$	0.751†	0.500†	0.730†	0.491†	0.436†	0.385†	0.384†
	$\text{MiniLM}_{\text{DPRCAT}}$	0.754*	0.503*	0.733*	0.492*	0.439*	0.389*	0.388*
DPR	$\text{MiniLM}_{\text{CAT}}$	0.734	0.506	0.717	0.495	0.411	0.357	0.354
	$\text{MiniLM}_{\text{BM25CAT}}$	0.757†	0.521†	0.734†	0.513†	0.462†	0.417†	0.416†
	$\text{MiniLM}_{\text{DPRCAT}}$	0.759*	0.527*	0.739*	0.517*	0.468*	0.421*	0.420*

Statistical significance was measured with a paired t-test ($p < 0.05$) with Bonferroni correction for multiple testing. The position of score injection is in the middle, between the query and the candidate document

† Indicates a statistically significant improvement of a cross-encoder with DPR score injection as text into the input (Cross-$\text{encoder}_{\text{BM25CAT}}$) over the same cross-encoder without BM25 score injection (Cross-$\text{encoder}_{\text{CAT}}$)

*Indicates a statistically significant improvement of a cross-encoder with DPR score injection as text into the input (Cross-$\text{encoder}_{\text{DPRCAT}}$) over the same cross-encoder without DPR score injection (Cross-$\text{encoder}_{\text{CAT}}$)

We observe that MiniLM yields results on par with BERT-Base for the MSMARCO-DEV dataset while demonstrating superior efficiency. Furthermore, our investigation reveals that across all four cross-encoders, incorporating DPR scores consistently improves the effectiveness more than injecting the BM25 score. This difference is clearly observable when comparing the results presented in Tables 2 and 8. However, it's worth noting that $\text{CE}_{\text{DPRCAT}}$ exhibits slower convergence during training when compared to $\text{CE}_{\text{BM25CAT}}$, a phenomenon that is discussed in more detail in Sect. 5.9.

Table 10 MRR@10 on MSMARCO-DEV per query type for comparing BERT-Base$_{BM25CAT}$ and BERT-Base$_{CAT}$ on different query types in full-text and exact-matching (only keeping query words) settings

Model	Input	ABBR	LOC	DESC	HUM	NUM	ENTY
# queries		9	493	1887	455	933	328
BERT-BaseCAT	Full text	0.574	0.477	0.397	0.435	0.361	0.399
BERT-BaseBM25CAT	Full text	**0.592**	**0.503**	**0.428**	**0.457**	**0.405**	**0.411**
BM25	Only query words	0.184	0.256	0.215	0.238	0.200	**0.221**
BERT-BaseCAT	Only query words	0.404	0.204	0.224	0.240	0.177	0.200
BERT-BaseBM25CAT	Only query words	**0.438**	**0.278**	**0.245**	**0.258**	**0.215**	0.216

Bold face indicates best performance

The position of score injection is in the middle, between the query and the candidate document

Fig. 4 Example query and two passages in the input of BERT$_{BM25CAT}$. The color of each word indicates the word-level attribution value according to Integrated Gradient (IG) [98], where red is positive, blue is negative, and white is neutral. We use the brightness of different colors to indicate the values of these gradients

Query [SEP] BM25 [SEP] Passage	Label	Model: Rank
[CLS] what is the shingles jab ? [SEP] 22 [SEP] the shingles vaccine . the vaccine , called zostavax , is given as a single injection under the skin (subcutaneously) . it can be given at any time in the year . unlike with the flu jab	R	BM25: 3 BERT$_{BM25CAT}$: 1 BERT$_{CAT}$: 104
[CLS] what is the shingles jab ? [SEP] 11 [SEP] shingle is a corruption of german schindle (schindel) meaning a roofing slate . shingles historically were called tiles and shingle was a term applied to wood shingles , as is still mostly the case outside the us [SEP]	N	BM25: 146 BERT$_{BM25CAT}$: 69 BERT$_{CAT}$: 1

5.7 Knowledge distillation

Table 9 demonstrates the significant impact of knowledge distillation on retrieval and ranking effectiveness. In this configuration, we consistently achieve higher effectiveness when training MiniLM for three variants: CE$_{CAT}$, CE$_{BM25CAT}$, and CE$_{DPRCAT}$ compared to Table 7, in which we do not use knowledge distillation. Notably, our observations indicate that through knowledge distillation, MiniLM$_{DPRCAT}$ attains even greater effectiveness compared to BERT-large$_{DPRCAT}$. For instance, when evaluating NDCG@10 on TREC DL'20, MiniLM$_{DPRCAT}$ achieves an impressive score of 0.759, surpassing BERT-Large$_{DPRCAT}$'s 0.757, as illustrated in Tables 8 and 9, respectively.

We consider this analysis to be of great importance from an industrial perspective, as knowledge distillation can significantly enhance the effectiveness of small cross-encoder re-rankers. This is important for keeping effeciency while increasing effectiveness in industry. Our findings also underscore the generalizability of CE$_{BM25CAT}$ and CE$_{DPRCAT}$ across various training configurations.

5.8 Analysis of the results

5.8.1 Query types

In order to analyze the effectiveness of BERT-base$_{CAT}$ and BERT-base$_{BM25CAT}$ across different types of questions, we classify questions based on the lexical answer type. We use the rule-based answer type classifier[10] inspired by Li and Roth [97] to extract answer types. We classify MSMARCO queries into 6 answer types: abbreviation, location, description, human, numerical and entity. 4105 queries have a valid answer type and at least one relevant passage in the top-1000. We perform our analysis in two different settings: normal (full-text) and exact-matching (keeping only query words and replacing non-query words with [*MASK*]). The average *MRR*@10 per query type is shown in Table 10. The table shows that BERT$_{BM25CAT}$ is more effective than BERT$_{CAT}$ consistently on all types of queries.

[10] https://github.com/superscriptjs/qtypes.

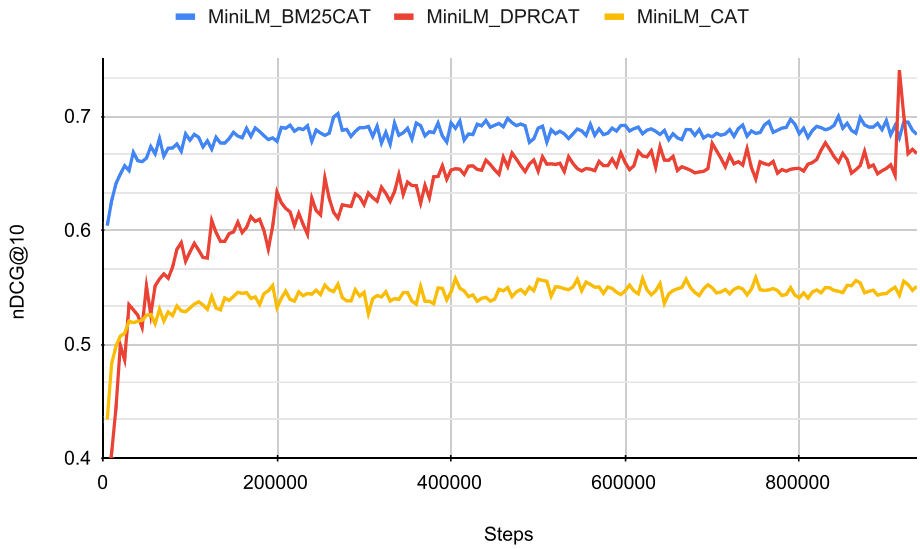

Fig. 5 Convergence analysis between $\text{MiniLM}_{\text{CAT}}$, $\text{MiniLM}_{\text{BM25CAT}}$, and $\text{MiniLM}_{\text{DPRCAT}}$. The nDCG@10 is reported at each step on the validation set during training

5.8.2 Qualitative analysis

We show a qualitative analysis of one particular case in Fig. 4 to analyze more in-depth what the effect of BM25 injection is and why it works. In the top row, while BERT_{CAT} mistakenly ranked the relevant passage at position 104, BM25 ranked that passage at position 3 and $\text{BERT}_{\text{BM25CAT}}$—apparently helped by BM25—ranked that relevant passage at position 1. In the bottom row, BERT_{CAT} mistakenly ranked the irrelevant passage at position 1 and informed by the low BM25 score, $\text{BERT}_{\text{BM25CAT}}$ ranked it much lower, at 69. In order to interpret the importance of the injected BM25 score in the input of $\text{CE}_{\text{BM25CAT}}$ and show its contributions to the matching score in comparison to other words in the query and passage, we use Integrated Gradient (IG) [98] which has been proven to be a stable and reliable interpretation method in many different applications including Information Retrieval [99–101].[11] On both rows of Fig. 4, we see that the BM25 score ('22' in the top row and '11' in the bottom row) is a highly attributed term in comparison to other terms. This shows that injecting the BM25 score assists $\text{BERT}_{\text{BM25CAT}}$ to identify relevant or non-relevant passages better than BERT_{CAT}.

As a more general analysis, we randomly sampled 100 queries from MSMARCO-DEV. For each query, we took the top-1000 passages retrieved by BM25, we fed all pairs of query and their corresponding retrieved passages (100*k* pairs) into $\text{BERT}_{\text{BM25CAT}}$, and computed the attribution scores over the input at the word-level. We ranked tokens based on their importance using the absolute value of their attribution score and found the mode of the rank of the BM25 token over all samples is 3. This shows that $\text{BERT}_{\text{BM25CAT}}$ highly attributes the BM25 token for ranking.

5.9 Convergence analysis

To assess the relation between the effectiveness of models and the training time required to reach the optimal weights, we present Fig. 5, which illustrates the nDCG@10 performance on the validation dataset throughout the training process. Our observations indicate that although $\text{CE}_{\text{DPRCAT}}$ exhibits slightly superior effectiveness compared to $\text{CE}_{\text{BM25CAT}}$, it requires a longer convergence time. This observation implies that when dealing with limited training data resources, choosing $\text{CE}_{\text{BM25CAT}}$ may yield a more effective cross-encoder re-ranker in a shorter training time as opposed to CE_{CAT} and $\text{CE}_{\text{DPRCAT}}$.

[11] We refer readers to [98] for a detailed explanation.

Table 11 Analyzing the relationship between entropy of normalized scores and effectiveness of $\text{MiniLM}_{\text{BM25CAT}}$ trained on them

Global maximum in Min-Max	Shannon entropy	nDCG@10
10	6.60	0.689
20	5.61	0.694
30	5.02	0.706
40	4.62	0.707
50	4.30	**0.711**
60	4.04	0.708
70	3.82	0.707
80	3.63	0.707
90	3.46	0.705
100	3.32	0.704

Bold face indicates best performance

Global minimum in Min–Max normalization is set to 0. The position of score injection is in the middle, between the query and the candidate document

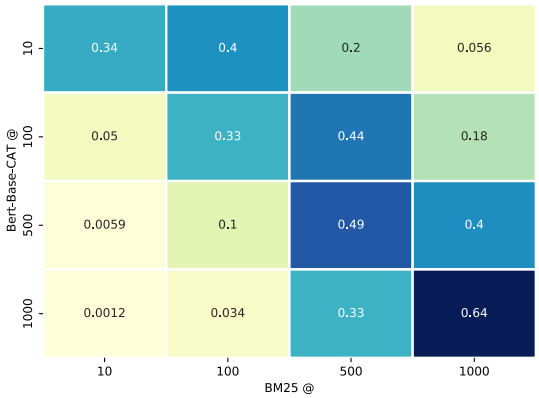

Fig. 6 Proportions of overlapping documents between $\text{BERT-Base}_{\text{CAT}}$ and BM25 ranking

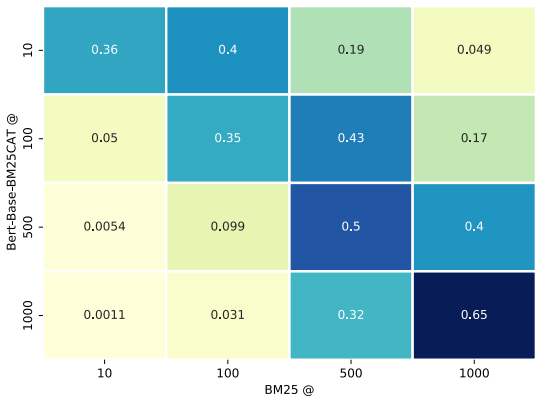

Fig. 7 Proportions of overlapping documents between $\text{BERT-Base}_{\text{BM25CAT}}$ and BM25 ranking

5.10 Entropy analysis

To delve deeper into the impact of score normalization variations within $\text{CE}_{\text{BM25CAT}}$, we conducted an in-depth assessment of $\text{MiniLM}_{\text{BM25CAT}}$'s effectiveness on TREC DL'19. This assessment involved manipulating the global maximum value in the min-max normalization process. Specifically, we set the global minimum to zero and incrementally raised the global maximum from 10 to 100 in steps of 10. We kept the global minimum as 0 and the reason for this choice is is explained previously in section 3.3. The results, presented in Table 11, reveal an interesting pattern.

As we increase the global maximum value, we observe a reduction in the entropy of the normalized scores, coinciding with an increase in effectiveness. However, a notable trend emerges: once the global maximum exceeds 50, there is a drop

in effectiveness. This suggests a potential trade-off between diminishing entropy in the scores and achieving enhanced effectiveness when injecting BM25 scores into $CE_{BM25CAT}$.

5.11 How do $CE_{BM25CAT}$ and BM25 rankings vary?

To interperet the difference between CE_{CAT} and $CE_{BM25CAT}$ in more-depth, inspired by Rau et al. [102], we plot the proportions of overlapping documents between BERT $Base_{CAT}$ and BM25, and BERT $Base_{BM25CAT}$ and BM25 in Fig. 6 and 7. Intuitively, each row indicates to what ratio documents stem from different rank-ranges. E.g., the top row can be read as the documents in rank 1-10 of the CE_{CAT} re-ranking originate 34% from rank 1 to 10, 40% from rank 11 to 100, 20% from rank 101 to 500 and 5.6% from rank 501 to 1000 in the initial BM25 ranking. In Fig. 7, we observe that that there is more similarity between ranking of $CE_{BM25CAT}$ and BM25 than CE_{CAT} and BM25. This could be a reason for the fact that $CE_{BM25CAT}$ is a more powerfull exact matcher.

6 Conclusion and future work

In this paper, we have proposed an efficient and effective way of combining first-stage retrievers and cross-encoder re-rankers. Prior research has primarily focused on the independent optimization of retrieval stages or simple linear combinations of scores from different models. Our approach deviates from these traditional methods by introducing a non-linear and continuous strategy for score integration, by injecting the first-stage retriever score as text in the input of the cross-encoder re-rankers. We find that the resulting models, $CE_{BM25CAT}$ and CE_{DPRCAT}, achieve a statistically significant improvement for all evaluated cross-encoders. Furthermore, the generalizability of our approach is demonstrated across various query types.

Our research builds upon the foundations of previous work that suggested the capability of BERT-based models in processing numeric data in textual representation. Our work provides a robust empirical example of the application of this ability of BERT-based models in information retrieval. We also found that injecting the BM25 or DPR relevance score in a knowledge distillation training setup can lead to statistically significant improvements.

Based on the experiments on injecting two different first-stage retrievers in two different training setups, we conclude that injecting the first-stage retriever relevance score is an impactful and straightforwardly available signal that leads to significant improvements in the effectiveness of cross-encoder re-rankers.

In conclusion, this work contributes to the information retrieval community by offering a refined, empirically validated method for the integration of first-stage retriever scores into cross-encoder re-rankers that might open new avenues for future research, potentially improving existing paradigms and encouraging a reevaluation of current retrieval practices. It provides a step forward in the development of more effective multi-stage retrieval systems. Future research, inspired by our findings, could further explore the implications of our approach in other contexts and for other types of data, e.g., score injection for first-stage dense passage retrievers.

Acknowledgements This work was supported by the EU Horizon 2020 ITN/ETN on Domain Specific Systems for Information Extraction and Retrieval (H2020-EU.1.3.1., ID: 860721).

This paper represents a substantial extension of our prior work, which was published at ECIR 2023 titled 'Injecting the BM25 Score as Text Improves BERT-Based Re-rankers' [1].

Author contributions Arian Askari: Methodology, Conceptualization, Implementation, Investigation, Writing—original draft Amin Abolghasemi: Writing—review and editing, Investigation Gabriella Pasi: Supervision, Writing—review and editing, Investigation Wessel Kraaij: Supervision, Writing—review and editing, Investigation Suzan Verberne: Supervision, Writing—review and editing, Investigation.

Funding This work was supported by the EU Horizon 2020 ITN/ETN on Domain Specific Systems for Information Extraction and Retrieval (H2020-EU.1.3.1., ID: 860721).

Data availability The datasets that have been used in this paper are publicly available thorough the following link https://github.com/microsoft/msmarco/blob/master/Datasets.md.

Declarations

Ethics approval and consent to participate Not applicable.

Competing interests The authors declare no competing interests.

References

1. Askari A, Abolghasemi A, Pasi G, Kraaij W, Verberne S. Injecting the BM25 score as text improves BERT-based re-rankers. In: Kamps J, Goeuriot L, Crestani F, Maistro M, Joho H, Davis B, Gurrin C, Kruschwitz U, Caputo A, editors. Advances in information retrieval. Cham: Springer; 2023. p. 66–83.
2. Robertson SE, Walker S. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In: SIGIR'94, Springer; 1994. p. 232–41.
3. Yan M, Li C, Wu C, Xia J, Wang W. IDST at TREC 2019 deep learning track: deep cascade ranking with generation-based document expansion and pre-trained language modeling. In: TREC; 2019.
4. Nogueira R, Cho K. Passage re-ranking with BERT. arXiv preprint arXiv:1901.04085; 2019.
5. Abolghasemi A, Verberne S, Azzopardi L. Improving BERT-based query-by-document retrieval with multi-task optimization. In: European Conference on Information Retrieval, Springer; 2022. p. 3–12.
6. Hofstätter S, Althammer S, Schröder M, Sertkan M, Hanbury A. Improving efficient neural ranking models with cross-architecture knowledge distillation. arXiv preprint arXiv:2010.02666; 2020.
7. Rau D, Kamps J. The role of complex NLP in transformers for text ranking. In: Proceedings of the 2022 ACM SIGIR International Conference on Theory of Information Retrieval; 2022. p. 153–60.
8. Anand M, Zhang J, Ding S, Xin J, Lin J. Serverless BM25 search and BERT reranking. In: DESIRES; 2021. p. 3–9.
9. Kamphuis C, de Vries AP, Boytsov L, Lin J. Which BM25 do you mean? A large-scale reproducibility study of scoring variants. In: European Conference on Information Retrieval, Springer; 2020. p. 28–34.
10. Gao L, Dai Z, Chen T, Fan Z, Durme BV, Callan J. Complement lexical retrieval model with semantic residual embeddings. In: European Conference on Information Retrieval, Springer; 2021. p. 146–60.
11. Lin J, Nogueira R, Yates A. Pretrained transformers for text ranking: Bert and beyond. Synth Lect Hum Lang Technol. 2021;14(4):1–325.
12. Rau D, Kamps J. How different are pre-trained transformers for text ranking? In: European Conference on Information Retrieval, Springer; 2022. pp. 207–14.
13. Salton G, McGill MJ. Introduction to modern information retrieval. New York: Mcgraw-hill; 1983.
14. Saracevic T. A review of an a framework for the thinking on the notion in information science. J Am Soc Inf Sci. 1975;26:321–43.
15. Wu S. Applying statistical principles to data fusion in information retrieval. Expert Syst Appl. 2009;36(2):2997–3006.
16. Wallace E, Wang Y, Li S, Singh S, Gardner M. Do NLP models know numbers? probing numeracy in embeddings. arXiv preprint arXiv:1909.07940; 2019.
17. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. In: Advances in Neural Information Processing Systems; 2017. p. 5998–6008.
18. Sanh V, Debut L, Chaumond J, Wolf T. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108; 2019.
19. Wang W, Wei F, Dong L, Bao H, Yang N, Zhou M. Minilm: deep self-attention distillation for task-agnostic compression of pre-trained transformers. Adv Neural Inf Proc Syst. 2020;33:5776–88.
20. Boualili L, Moreno JG, Boughanem M. Markedbert: Integrating traditional IR cues in pre-trained language models for passage retrieval. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval; 2020. p. 1977–80.
21. Boualili L, Moreno JG, Boughanem M. Highlighting exact matching via marking strategies for ad hoc document ranking with pretrained contextualized language models. Inf Retr J. 2022;25:1–47.
22. Nogueira R, Yang W, Lin J, Cho K. Document expansion by query prediction. arXiv preprint arXiv:1904.08375; 2019.
23. Al-Hajj M, Jarrar M. Arabglossbert: fine-tuning BERT on context-gloss pairs for wsd. arXiv preprint arXiv:2205.09685; 2022.
24. Li L, Dai Y, Tang D, Feng Z, Zhou C, Qiu X, Xu Z, Shi S. Markbert: marking word boundaries improves Chinese BERT. arXiv preprint arXiv:2203.06378; 2022.
25. Askari A, Aliannejadi M, Abolghasemi A, Kanoulas E, Verberne S. Closer: conversational legal longformer with expertise-aware passage response ranker for long contexts. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. CIKM '23, Association for Computing Machinery, New York, NY, USA; 2023. p. 25–35. https://doi.org/10.1145/3583780.3614812.
26. Han J, Hong T, Kim B, Ko Y, Seo J. Fine-grained post-training for improving retrieval-based dialogue systems. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2021. p. 1549–58.
27. Askari A, Yang Z, Ren Z, Verberne S. Answer retrieval in legal community question answering. In: European Conference on Information Retrieval, Springer; 2024. p. 477–85.
28. Gretkowski A, Wiśniewski D, Ławrynowicz A. Should we afford affordances? Injecting conceptnet knowledge into BERT-based models to improve commonsense reasoning ability. In: Corcho O, Hollink L, Kutz O, Troquard N, Ekaputra FJ, editors. Knowledge engineering and knowledge management. Cham: Springer; 2022. p. 97–104.
29. Thawani A, Pujara J, Szekely PA, Ilievski F. Representing numbers in NLP: a survey and a vision. arXiv preprint arXiv:2103.13136; 2021.

30. Geva M, Gupta A, Berant J. Injecting numerical reasoning skills into language models. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics; 2020. p. 946–58. https://www.aclweb.org/anthology/2020.acl-main.89. Accessed 01 Aug 2023
31. Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, Agarwal S, Herbert-Voss A, Krueger G, Henighan T, Child R, Ramesh A, Ziegler DM, Wu J, Winter C, Hesse C, Chen M, Sigler E, Litwin M, Gray S, Chess B, Clark J, Berner C, McCandlish S, Radford A, Sutskever I, Amodei D. Language models are few-shot learners. Adv Neural Inf Process Syst. 2020;33:1877–901.
32. Naik A, Ravichander A, Rose C, Hovy E. Exploring numeracy in word embeddings. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy; 2019. p. 3374–80. https://doi.org/10.18653/v1/P19-1329. https://www.aclweb.org/anthology/P19-1329. Accessed 01 Aug 2023
33. Johnson D, Mak D, Barker A, Loessberg-Zahl L. Probing for multilingual numerical understanding in transformer-based language models. In: Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, Association for Computational Linguistics; 2020. p. 184–92. https://www.aclweb.org/anthology/2020.blackboxnlp-1.18. Accessed 01 Aug 2023
34. Amini A, Gabriel S, Lin S, Koncel-Kedziorski R, Choi Y, Hajishirzi H. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota; 2019. p. 2357–67. https://doi.org/10.18653/v1/N19-1245. https://www.aclweb.org/anthology/N19-1245. Accessed 01 Aug 2023
35. Saxton D, Grefenstette E, Hill F, Kohli P. Analysing mathematical reasoning abilities of neural models. In: International Conference on Learning Representations; 2019. https://openreview.net/forum?id=H1gR5iR5FX. Accessed 01 Aug 2023
36. Roy S, Roth D. Solving general arithmetic word problems. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Lisbon, Portugal; 2015. p. 1743–52. https://doi.org/10.18653/v1/D15-1202. https://www.aclweb.org/anthology/D15-1202. Accessed 01 Aug 2023
37. Hendrycks D, Burns C, Kadavath S, Arora A, Basart S, Tang E, Song D, Steinhardt J. Measuring mathematical problem solving with the math dataset. arXiv preprint arXiv:2103.03874; 2021.
38. Mishra S, Mitra A, Varshney N, Sachdeva B, Baral C. Towards question format independent numerical reasoning: a set of prerequisite tasks; 2020.
39. Lin BY, Lee S, Khanna R, Ren X. Birds have four legs?! NumerSense: probing Numerical Commonsense Knowledge of Pre-Trained Language Models. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online; 2020. p. 6862–8. https://doi.org/10.18653/v1/2020.emnlp-main.557. https://www.aclweb.org/anthology/2020.emnlp-main.557. Accessed 01 Aug 2023
40. Bullard SE, Fein D, Gleeson MK, Tischer N, Mapou RL, Kaplan E. The Biber cognitive estimation test. Arch Clin Neuropsychol. 2004;19(6):835–46.
41. Forbes M, Choi Y. Verb physics: Relative physical knowledge of actions and objects. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vancouver, Canada; 2017. p. 266–76. https://doi.org/10.18653/v1/P17-1025. https://www.aclweb.org/anthology/P17-1025. Accessed 01 Aug 2023
42. Elazar Y, Mahabal A, Ramachandran D, Bedrax-Weiss T, Roth D. How large are lions? Inducing distributions over quantitative attributes. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy; 2019. p. 3973–83. https://doi.org/10.18653/v1/P19-1388. https://www.aclweb.org/anthology/P19-1388. Accessed 01 Aug 2023
43. Zhang X, Ramachandran D, Tenney I, Elazar Y, Roth D. Do language embeddings capture scales? In: Findings of the Association for Computational Linguistics: EMNLP 2020, Association for Computational Linguistics; 2020. p. 4889–96. https://www.aclweb.org/anthology/2020.findings-emnlp.439. Accessed 01 Aug 2023
44. Zhou B, Ning Q, Khashabi D, Roth D. Temporal common sense acquisition with minimal supervision. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics; 2020. p. 7579–89. https://doi.org/10.18653/v1/2020.acl-main.678. https://www.aclweb.org/anthology/2020.acl-main.678. Accessed 01 Aug 2023
45. Dubey A, Kumar L, Somani A, Joshi A, Bhattacharyya P. "When numbers matter!": detecting sarcasm in numerical portions of text. In: Proceedings of the Tenth Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, Association for Computational Linguistics, Minneapolis, USA; 2019. p. 72–80. https://doi.org/10.18653/v1/W19-1309. https://www.aclweb.org/anthology/W19-1309. Accessed 01 Aug 2023
46. Chen C-C, Huang H-H, Chen H-H. Numclaim: investor's fine-grained claim detection. In: Proceedings of the 29th ACM International Conference on Information and Knowledge Management. CIKM '20, Association for Computing Machinery, New York, NY, USA; 2020. p. 1973–6. https://doi.org/10.1145/3340531.3412100. Accessed 01 Aug 2023
47. Zhang X, Ramachandran D, Tenney I, Elazar Y, Roth D. Do language embeddings capture scales? arXiv preprint arXiv:2010.05345; 2020.
48. Berg-Kirkpatrick T, Spokoyny D. An empirical investigation of contextualized number prediction. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2020. p. 4754–64.
49. Muffo M, Cocco A, Bertino E. Evaluating transformer language models on arithmetic operations using number decomposition. In: Proceedings of the Thirteenth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France; 2022. p. 291–297. https://aclanthology.org/2022.lrec-1.30. Accessed 01 Aug 2023
50. Johnson D, Mak D, Barker D, Loessberg-Zahl L. Probing for multilingual numerical understanding in transformer-based language models. arXiv preprint arXiv:2010.06666; 2020.
51. Geva M, Gupta A, Berant J. Injecting numerical reasoning skills into language models. arXiv preprint arXiv:2004.04487; 2020.
52. Gu K, Budhkar A. A package for learning on tabular and text data with transformers. In: Proceedings of the Third Workshop on Multimodal Artificial Intelligence, Association for Computational Linguistics, Mexico City, Mexico; 2021. p. 69–73. https://doi.org/10.18653/v1/2021.maiworkshop-1.10. https://www.aclweb.org/anthology/2021.maiworkshop-1.10. Accessed 01 Aug 2023
53. Upadhyay R, Askari A, Pasi G, Viviani M. Enhancing documents with multidimensional relevance statements in cross-encoder re-ranking. arXiv preprint arXiv:2306.10979; 2023.

54. Bartell BT, Cottrell GW, Belew RK. Automatic combination of multiple ranked retrieval systems. In: SIGIR'94, Springer; 1994. p. 173–81.
55. Askari A, Verberne S, Pasi G. Expert finding in legal community question answering. In: Hagen M, Verberne S, Macdonald C, Seifert C, Balog K, Nørvåg K, Setty V, editors. Advances in information retrieval. Cham: Springer; 2022. p. 22–30.
56. Askari A, Verberne S. Combining lexical and neural retrieval with longformer-based summarization for effective case law retrieval. In: Proceedings of the Second International Conference on Design of Experimental Search & Information REtrieval Systems, CEUR; 2021. p. 162–70.
57. Bassani E, Romelli L. ranx.fuse: A python library for metasearch. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management. CIKM '22, Association for Computing Machinery, New York, NY, USA; 2022. p. 4808–12. https://doi.org/10.1145/3511808.3557207.
58. Fox E, Shaw J. Combination of multiple searches. NIST Special Publication SP; 1994. p. 243.
59. Lee JH. Analyses of multiple evidence combination. In: Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval; 1997. p. 267–76.
60. Wu S, Crestani F. Data fusion with estimated weights. In: Proceedings of the Eleventh International Conference on Information and Knowledge Management; 2002. p. 648–651.
61. Bailey P, Moffat A, Scholer F, Thomas P. Retrieval consistency in the presence of query variations. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval; 2017. p. 395–404.
62. Cormack GV, Clarke CL, Buettcher S. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval; 2009. p. 758–9.
63. Mourão A, Martins F, Magalhaes J. Multimodal medical information retrieval with unsupervised rank fusion. Comput Med Imag Grap. 2015;39:35–45.
64. Aslam JA, Montague M. Models for metasearch. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval; 2001. p. 276–84.
65. Lillis D, Toolan F, Collier R, Dunnion J. Probfuse: a probabilistic approach to data fusion. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval; 2006. p. 139–46.
66. Lillis D, Toolan F, Collier R, Dunnion J. Extending probabilistic data fusion using sliding windows. In: Advances in Information Retrieval: 30th European Conference on IR Research, ECIR 2008, Glasgow, UK, March 30-April 3, 2008. Proceedings 30, Springer; 2008. p. 358–69.
67. Lillis D, Zhang L, Toolan F, Collier RW, Leonard D, Dunnion J. Estimating probabilities for effective data fusion. In: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval; 2010. p. 347–54.
68. Montague M, Aslam JA. Condorcet fusion for improved retrieval. In: Proceedings of the Eleventh International Conference on Information and Knowledge Management; 2002. p. 538–48.
69. Zhang Y, Hu C, Liu Y, Fang H, Lin J. Learning to rank in the age of muppets: effectiveness–efficiency tradeoffs in multi-stage ranking. In: Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing; 2021. p. 64–73.
70. Wang S, Zhuang S, Zuccon G. Bert-based dense retrievers require interpolation with bm25 for effective passage retrieval. In: Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval. ICTIR '21, Association for Computing Machinery, New York, NY, USA; 2021. p. 317–24. https://doi.org/10.1145/3471158.3472233.
71. Abolghasemi A, Askari A, Verberne S. On the interpolation of contextualized term-based ranking with BM25 for query-by-example retrieval. In: Proceedings of the 2022 ACM SIGIR International Conference on Theory of Information Retrieval. ICTIR '22, Association for Computing Machinery, New York, NY, USA; 2022. p. 161–70. https://doi.org/10.1145/3539813.3545133.
72. Althammer S, Askari A, Verberne S, Hanbury A. DoSSIER@ COLIEE 2021: leveraging dense retrieval and summarization-based re-ranking for case law retrieval. arXiv preprint arXiv:2108.03937; 2021.
73. Lin S-C, Yang J-H, Lin J. Distilling dense representations for ranking using tightly-coupled teachers. arXiv preprint arXiv:2010.11386; 2020.
74. Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531; 2015.
75. Zhao WX, Liu J, Ren R, Wen J-R. Dense text retrieval based on pretrained language models: a survey. ACM Trans Inf Syst. 2024;42(4):1–60.
76. Robertson S, Zaragoza H, et al. The probabilistic relevance framework: Bm25 and beyond. Found Trends® Inf Retr. 2009;3(4):333–89.
77. Karpukhin V, Oguz B, Min S, Lewis P, Wu L, Edunov S, Chen D, Yih W-T. Dense passage retrieval for open-domain question answering. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics; 2020. pp. 6769–81. https://doi.org/10.18653/v1/2020.emnlp-main.550. https://www.aclweb.org/anthology/2020.emnlp-main.550. Accessed 01 Aug 2023
78. Lin J, Ma X, Lin S-C, Yang J-H, Pradeep R, Nogueira R. Pyserini: BM25 baseline for MS MARCO document retrieval; 2021. https://github.com/castorini/pyserini/blob/master/docs/experiments-msmarco-doc.md. Accessed 01 Aug 2023
79. Lin J, Ma X, Lin S-C, Yang J-H, Pradeep R, Nogueira R. Pyserini: a Python toolkit for reproducible information retrieval research with sparse and dense representations. In: Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021); 2021. p. 2356–62.
80. MacAvaney S, Yates A, Cohan A, Goharian N. CEDR: Contextualized embeddings for document ranking. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval; 2019. p. 1101–4.
81. Yilmaz ZA, Yang W, Zhang H, Lin J. Cross-domain modeling of sentence-level evidence for document retrieval. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP); 2019. p. 3490–6.
82. Michael N, Diego C, Joshua P, LP B. Learning to rank; 2022. https://solr.apache.org/guide/solr/latest/query-guide/learning-to-rank.html#feature-engineering. Accessed 01 Aug 2023
83. Zhang X, Yates A, Lin J. Comparing score aggregation approaches for document retrieval with pretrained transformers. In: Hiemstra D, Moens M-F, Mothe J, Perego R, Potthast M, Sebastiani F, editors. Advances in information retrieval. Cham: Springer; 2021. p. 150–63.
84. Nguyen T, Rosenberg M, Song X, Gao J, Tiwary S, Majumder R, Deng L. Ms marco: a human generated machine reading comprehension dataset. In: CoCo@ NIPs; 2016.

85. Craswell N, Mitra B, Yilmaz E, Campos D, Voorhees EM. Overview of the TREC 2019 deep learning track. arXiv preprint arXiv:2003.07820; 2020.
86. Craswell N, Mitra B, Yilmaz E, Campos D. Overview of the TREC 2020 deep learning track. arXiv preprint arXiv:2102.07662; 2021.
87. Khattab O, Zaharia M. Colbert: efficient and effective passage search via contextualized late interaction over BERT. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval; 2020. p. 39–48.
88. MacAvaney S, Nardini FM, Perego R, Tonellotto N, Goharian N, Frieder O. Expansion via prediction of importance with contextualization. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval; 2020. p. 1573–6.
89. Zhuang S, Zuccon G. Tilde: Term independent likelihood model for passage re-ranking. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval; 2021. p. 1483–92.
90. Zhuang S, Li H, Zuccon G. Deep query likelihood model for information retrieval. In: European Conference on Information Retrieval, Springer; 2021. p. 463–70.
91. Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, Cistac P, Rault T, Louf R, Funtowicz M, et al. Huggingface's transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771; 2019.
92. Reimers N, Gurevych I. Sentence-BERT: sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, arXiv:1908.10084; 2019.
93. Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, Lin Z, Desmaison A, Antiga L, Lerer A. Automatic differentiation in pytorch; 2017.
94. Kingma DP, Ba J. Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980; 2014.
95. Zhang Z, Sabuncu M. Generalized cross entropy loss for training deep neural networks with noisy labels. Adv Neural Inf Process Syst. 2018;31.
96. Craswell N, Mitra B, Yilmaz E, Campos D, Lin J. Ms marco: benchmarking ranking models in the large-data regime. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval; 2021. p. 1566–76.
97. Li X, Roth D. Learning question classifiers. In: COLING 2002: the 19th International Conference on Computational Linguistics; 2002.
98. Sundararajan M, Taly A, Yan Q. Axiomatic attribution for deep networks. In: International Conference on Machine Learning, PMLR; 2017. p. 3319–28.
99. Zhan J, Mao J, Liu Y, Zhang M, Ma S. An analysis of BERT in document ranking. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020; pp. 1941–4.
100. Chen L, Lan Y, Pang L, Guo J, Cheng X. Toward the understanding of deep text matching models for information retrieval. arXiv preprint arXiv:2108.07081; 2021.
101. Zhan J, Mao J, Liu Y, Guo J, Zhang M, Ma S. Interpreting dense retrieval as mixture of topics. arXiv preprint arXiv:2111.13957; 2021.
102. Rau D, Kamps J. How different are pre-trained transformers for text ranking? In: Hagen M, Verberne S, Macdonald C, Seifert C, Balog K, Nørvåg K, Setty V, editors. Advances in information retrieval. Cham: Springer; 2022. p. 207–14.